

**Amendments to the Drawings:**

In accordance with the Examiners objections to the drawings in the Office Action, the attached sheets of drawings includes changes to Table 1 and Table 2. These sheet, which includes Fig. 10 and Fig. 11, replaces the original sheets including Table 1 and Table 2.

Attachment: Replacement Sheets

## **REMARKS**

### **Priority**

Applicant submits herewith the certified copy of the priority application.

### **Drawings**

Applicant submits herewith substitute drawing sheets for those sheets including “Table 1” and “Table 2.” In addition, the Brief Description of the Drawings” has been amended to conform to the substitute drawings sheets.

### **Status of the Claims**

Applicant submits herewith claims that are in a format similar to that originally present in this application (i.e., claims 1-38, present before submission of the preliminary amendment). The new claims are numbered 75-113. The new claims are similar to the originally-filed claims, but with some amendments/differences that are discussed throughout this response, in the appropriate sections.

### **Claim Rejections – Statutory Subject Matter**

The Examiner contends that the subject matter recited in the claims is non-statutory subject matter. As discussed above, new claims have been provided. The method claims are directed to a computer-implemented method. The system claims include a combination of statutory article of manufacture elements.

It is thus respectfully submitted that the claims recite statutory subject matter.

### **Claim Rejections -- Enablement**

The claims are rejected as not having enabling support in the specification. It is respectfully submitted that the new claims are, in fact, enabled by the specification.

In particular, the elements recited in the claims are described such that one of ordinary skill in the art may practice these elements. In particular, the claims now recite more specific steps of the method, and these steps are clearly described in the specification and shown in the drawings. Similarly, the elements of the system claims are clearly described in the specification and shown in the drawings.

### **Claims Rejections – Lack of Antecedent Basis**

Claims 55, 56, 73 and 74 are rejected as having terms for which antecedent basis has not been provided. The rejected claims 55, 56, 73 and 74 are most similar to new claims 94 and 113.

The consistent reference to C++ class in the preceding claims from which claim 94 and 113 depend clarifies the antecedent basis and removes the reason for objection.

#### **Claim Rejections – Anticipation (Comair et al)**

Claims 39 and 57 are rejected as being anticipated by Comair et al. It is respectfully submitted that the subject matter of the new claims 75-113 are patentably distinct over Comair et al.

Pub simply, the subject matter of the new claims is distinct from that disclosed by Comair et al. because the program disclosed by Comair, although used in modeling, already contains all the numeric data and/or syntactic keywords and/or defined algorithmic expressions in a compiled form.

This interpretation is confirmed with reference to Figure 17 of Comair et al. More specifically, item 628 the Builder function is described at column 14 lines 43 to 45. There, it is stated that the “Builder 628 generates and edits runtime code, performs compilation and link environment setup, and performs executable builds”.

Further in column 14 at lines 57 to 60 of the Comair citation, “the virtual device interface 634 may allow the executable code generated by builder 628 to be executed on a number of target platforms”.

Comair et al. discloses modelling objects using a generic plan that specifies the behavior of the object and how it interacts with objects. At column 2 lines 65 to 67, it is stated that “the abstract data structures/methods provided in accordance with the present invention are general enough to suit tool-kit and run-time software ...,” from which it is known that there must be a compiling step so as assimilate abstract and physical components of the simulation entities before the final product is processed for use by the user.

It is also of note that the Comair et al. disclosure does not describe any user defined algorithms being implemented for any of the entities described. Rather, it describes pre-determined algorithmic expressions that need to be incorporated into the code by compiling all the expressions and data described above.

The independent claims of the application, by contrast, include the execution of the data from the data file with the compiled generic description, without a need to recompile the generic description, so as to represent on a computer the behavior of a system.

It is therefore respectfully submitted that all of the claims recite subject matter that is patentably distinct from the disclosure of Comair.

**Claim Rejections – Obviousness (Paterson et al)**

Claim 57 is rejected as being obvious in view of Paterson et al. The rejection is discussed in the context of new claim 95.

In particular, the Paterson et al. specification refers to the user selecting from pre-defined parameters and objects to build up their model. Paterson et al. discloses that the selections are made from pre-compiled code options and that all the variables required have been anticipated and pre-programmed, precompiled and then made available to the user to select. There is no ability to run the pre-compiled program with user defined algorithms to allow for a requirement that was not anticipated -- and therefore not pre-programmed and pre-compiled -- without recompiling the code.

The only reference in the Paterson disclosure to a user defined equation is to a diagrammatic representation which is not run on the fly, as is possible using the subject matter recited in new claim 95. Furthermore, there is no suggestion to modify the Paterson disclosure to include this feature.

**Claim Rejections – Obviousness (Comair, in view of Stroustrup and Banks)**

The Examiner contends that “the C++ programming language” by Stroustrup and the “Handbook of Simulation” edited by Jerry Banks (Stroustrup and Banks respectively) is properly read in conjunction with the Comair patent and, furthermore, that claims 42-44, 46, 48-51, 55-56, 60-62, 64, 66-69 and 73-74 are rendered obvious.

The Examiner cites, in particular, that in the C++ language there exist a known function described as “templating,” which as cited “provides direct support for generic programming” and that specialization is a concept supported by templates. The Examiner also states that in light of this knowledge, it is obvious to “provide a specialized function that replaces an abstracted or generic function in a base template”.

The Examiner further expresses the combination in light of the invention of Comair when implemented in object orientated C++ language, including templates as taught by Stroustrup and adhering to the principles of modelling and simulation as taught by Banks. The Examiner rejects the claims because they are obvious in light of this combination.

Applicant respectfully submits that the cited combination is not applicable to the new claims. In particular, while perhaps it would be legitimate to cite C++ language and simulation if the recited subject matter were such to require the generic model to be recompiled. However, as discussed previously, the independent claims of the application include the execution of the data from the data file with the compiled generic description, without a need to recompile the generic description. By contrast, all of the examples cited by the Examiner relate to an implementation at a programming language level, and the resulting code still needs to be compiled using a compiler program to produce executable code for use by the simulation user.

To express the point in another way, the C++ templates described would need to be modified in the generic model in order to modify the system behavior, and that modification can only be achieved by modifying the source code and re-compiling the modified source code before it can be used by the simulation user.

The recited subject matter is distinct in that the generic description is precompiled and does not need to be recompiled "to include the data obtained from the data files so as to calculate the behavior of the said system" when executing the compiled generic description using an interval.

### **CONCLUSION**

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP



Alan S. Hodes  
Reg. No. 38,185

P.O. Box 70250  
Oakland, CA 94612-0250  
(650) 961-8300